

(X)



(12) **EUROPEAN PATENT APPLICATION**

(43) Date of publication:
25.06.2003 Bulletin 2003/26

(51) Int Cl.7: **H04M 1/73, H04B 1/16**

(21) Application number: **02255670.8**

(22) Date of filing: **14.08.2002**

(84) Designated Contracting States:
**AT BE BG CH CY CZ DE DK EE ES FI FR GB GR
IE IT LI LU MC NL PT SE SK TR**
Designated Extension States:
AL LT LV MK RO SI

- **Taniguchi, Shoji, c/o Fujitsu Limited**
Kawasaki-shi, Kanagawa 211-8588 (JP)
- **Kanasugi, Masami, c/o Fujitsu Limited**
Kawasaki-shi, Kanagawa 211-8588 (JP)

(30) Priority: **18.12.2001 JP 2001384256**

(71) Applicant: **FUJITSU LIMITED**
Kawasaki-shi, Kanagawa 211-8588 (JP)

(74) Representative: **Hitching, Peter Matthew et al**
Haseltine Lake & Co.,
Imperial House,
15-19 Kingsway
London WC2B 6UD (GB)

(72) Inventors:
• **Kuroiwa, Koichi, c/o Fujitsu Limited**
Kawasaki-shi, Kanagawa 211-8588 (JP)

(54) **Semiconductor device, portable remote terminal unit and intermittent receiving method**

(57) A semiconductor device is disclosed in which consumption of power attendant on a leakage current in a block (2) where the power is turned on and off intermittently is reduced. In the semiconductor device according to the present invention, an error check necessity judgment circuit (4) judges on notification from an error check necessity notification circuit (6) located in a block (2) where the power is not disconnected whether an error check must be made, and an error check execution circuit (5) makes an error check on data loaded

from an external memory at the time of a boot in accordance with the judgment of the error check necessity judgment circuit (4). An error check cannot completely be omitted. Therefore, to ensure the reliability of a system, an error check is forcedly made once whenever a boot is performed a number of times set by an error check interval setting circuit (7). This shortens intermittent operation time by time taken to make an error check, resulting in a reduction in consumption of power attendant on a useless leakage current.

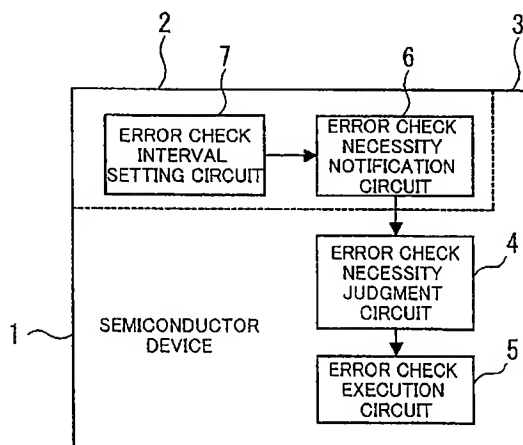


FIG. 1

Description

[0001] This application is based upon and claims priority of Japanese Patent Application No.2001-384256, filed on December 18, 2001, the contents being incorporated herein by reference.

[0002] This invention relates to a semiconductor device and, more particularly, to a semiconductor device such as a large-scale integration (LSI) device in a portable remote terminal unit used in a direct-sequence spread spectrum communication system, which realizes a reduction in consumption of power by periodically turning internal power on and off in a standby state at nonuse time.

[0003] In recent years microfabrication technologies have been adopted in the process for producing LSI and wiring width is approaching 0.10 μm . These microfabrication technologies increase the packaging density of transistors, but threshold voltages LSI needs to operate must be lowered. This causes the substantial problem of a leakage (quiescent) current between transistors increasing at standby time. A portable remote terminal unit, such as a cellular telephone, which is driven by a battery is always in a standby state even at nonuse time. Therefore, an increase in this leakage current will shorten nonstop telephone call time and standby time. This is a very serious problem from a practical viewpoint.

[0004] Conventionally, the power has been disconnected from sections which do not need to operate in a standby state to prevent consumption of power from increasing due to such a leakage current. In addition, in LSI, the power has been disconnected from blocks which do not need to operate to prevent consumption of power from increasing due to such a leakage current.

[0005] When the power is disconnected from what is called processors, such as a digital signal processor (DSP) and central processing unit (CPU), included in LSI, all the data the processors were processing until then will be lost. It therefore is necessary to save it in a nonvolatile memory for the reapplication of the power. That is to say, the state before the power being turned off is saved in a memory outside the LSI where the power is not disconnected. When power is reapplied, data saved in this memory will be restored.

[0006] The boot process of loading a program from an external memory is performed by a program in a processor each time the power is turned on.

[0007] Fig. 10 is a view for describing the timing with which a boot process is performed.

[0008] Each time a boot process is performed, an error detection process called checksum is performed after a program being loaded from an external memory to judge whether the program was read normally.

[0009] For example, with a cellular telephone in a standby state, power is applied only during intermittent receiving to reduce consumption of power. That is to say, the power is disconnected when intermittent receiving is not performed. At the time when a boot is per-

formed for this intermittent receiving, error detection is performed.

[0010] This intermittent receiving is the operation of checking whether the owner of a cellular telephone had a telephone call from a base station. A cellular telephone is put into a receiving state in a cycle between, for example, one and two seconds. If there is no incoming call, then the power is disconnected.

[0011] Fig. 11 is a view for describing an intermittent receiving cycle and checksum time.

[0012] With a cellular telephone, a boot interval is the same as an intermittent receiving cycle in a standby state and is usually 1.28 or 2.56 seconds. If a cellular telephone is in a standby state, power is applied once every intermittent receiving cycle. This intermittent receiving cycle includes program load time and checksum execution time.

[0013] With a cellular telephone and the like which can be in a standby state for hundreds of hours, however, the percentage of accumulated time for which checksum is performed to the total time for which the power should be in the ON state will be high if the checksum is performed at the above boot intervals. A system cannot operate while checksum is being performed, so there are many blocks where a useless leakage current flows. This means that a battery is used wastefully. Moreover, a program is loaded every time, so a useless leakage current also flows during load time.

[0014] It is desirable to provide a semiconductor device in which consumption of power attendant on a leakage current is reduced by shortening time for which a system cannot operate.

[0015] According to an aspect of the present invention there is provided a semiconductor device with an intermittent operation block where application and disconnection of the power are performed periodically and repeatedly. This semiconductor device comprises an error check necessity judgment circuit for judging whether an error check must be made on data for booting loaded from an external memory by a boot process performed at the time of power being applied and an error check execution circuit for making an error check on the data for booting in the case of the error check necessity judgment circuit judging that an error check must be made.

[0016] The above and other objects, features and advantages of the present invention will become apparent from the following description when taken in conjunction with the accompanying drawings which illustrate preferred embodiments of the present invention by way of example.

[0017] Fig. 1 is a view showing the theoretical structure of a semiconductor device according to the present invention.

[0018] Fig. 2 is a view showing the structure of the feature of a system necessary for intermittent receiving operation.

[0019] Fig. 3 is a view showing the structure of checksum control.

[0020] Fig. 4 is a flow chart showing the sequence of processes performed regarding boot programs.

[0021] Fig. 5 is a view showing the difference between time for which the power is in the ON state in the case of checksum being performed and time for which the power is in the ON state in the case of checksum not being performed.

[0022] Fig. 6 is a view showing the details of a checksum control register.

[0023] Fig. 7 is a flow chart showing the sequence of processes performed regarding a boot program which performs checksum forcedly and periodically.

[0024] Fig. 8 is a view showing an example of dividing a program storage area in a memory.

[0025] Fig. 9 is a flow chart showing the sequence of processes performed regarding divided boot programs.

[0026] Fig. 10 is a view for describing the timing with which a boot process is performed.

[0027] Fig. 11 is a view for describing an intermittent receiving cycle and checksum time.

[0028] An overview of the present invention will now be given with reference to the drawings.

[0029] Fig. 1 is a view showing the theoretical structure of a semiconductor device according to the present invention.

[0030] A semiconductor device 1 according to the present invention comprises a block 2 where power is always applied and an intermittent operation block 3 where application and disconnection of the power are performed periodically and repeatedly. The intermittent operation block 3 includes an error check necessity judgment circuit 4 and error check execution circuit 5. The block 2 includes an error check necessity notification circuit 6 and error check interval setting circuit 7.

[0031] The error check necessity judgment circuit 4 judges whether an error check must be made on data loaded from an external memory by a boot process performed at the time when power is applied. The error check execution circuit 5 makes an error check for confirming whether data was loaded normally in the case of the error check necessity judgment circuit 4 judging that an error check must be made.

[0032] The error check necessity notification circuit 6 located in the block 2 where the power is not disconnected notifies the error check necessity judgment circuit 4 of instructions about whether an error check must be made. Only when the number of times a boot was performed reaches a certain number, the error check interval setting circuit 7 causes the error check necessity notification circuit 6 to notify the error check necessity judgment circuit 4 that an error check must be made.

[0033] In the semiconductor device 1 having the above structure, the error check necessity notification circuit 6 and error check interval setting circuit 7 are located in the block 2 where the power is not disconnected. Usually the error check necessity notification circuit 6 notifies the error check necessity judgment circuit 4 that an error check is unnecessary. The error check in-

terval setting circuit 7 counts the number of times power is applied to the intermittent operation block 3. When the number of times power is applied to the intermittent operation block 3, that is to say, the number of times a boot process is performed as a result of it reaches a certain number, the error check interval setting circuit 7 changes the value set for the error check necessity notification circuit 6 so that an error check will be made.

[0034] When power is applied to the intermittent operation block 3, data regarding a program for booting is loaded first from an external memory. The error check necessity judgment circuit 4 will be notified at this time by the error check necessity notification circuit 6 whether to make an error check, so the error check necessity judgment circuit 4 judges the contents of the notification. If an error check is unnecessary, then an error check process on the data loaded will be omitted.

[0035] When the number of times a boot is performed reaches a certain number after intermittent operation being repeated, the error check interval setting circuit 7 changes the value set for the error check necessity notification circuit 6 so that an error check will be made, and the error check necessity notification circuit 6 notifies the error check necessity judgment circuit 4. The error check necessity judgment circuit 4 therefore judges that an error check must be made. In this case, an error check will be made on the data regarding the program for booting after the data is loaded from the external memory.

[0036] As described above, the semiconductor device 1 according to the present invention can determine at the time of a boot whether to make an error check on data loaded from an external memory. As a result, there is no need to make an error check every time on data loaded each time a boot is performed. This shortens intermittent operation time by time taken to make these error checks, resulting in a reduction in consumption of power attendant on a useless leakage current.

[0037] In the above example, the block 2 where power is always applied is within the semiconductor device 1. The function of the block 2 may be located outside the semiconductor device 1. In this case, the semiconductor device 1 includes an input terminal for accepting data indicative of whether an error check must be made, and the error check necessity judgment circuit 4 will judge on data indicative of whether an error check must be made, which is accepted at this input terminal, whether an error check must be made.

[0038] Now, an embodiment of the present invention applied to LSI for cellular telephones which perform intermittent receiving will be described in detail.

[0039] Fig. 2 is a view showing the structure of the feature of a system necessary for intermittent receiving operation.

[0040] LSI 10 includes a power on block 11 where the power is not turned off and a power on/off block 12 where the power is turned on and off to an intermittent receiving cycle. The power on/off block 12 includes a

logic circuit 13 including a modem section for modulating and demodulating signals, such as voice, and a CO-DEC section for performing a scramble process and error correction process, a DSP 14 for controlling the logic circuit 13, a CPU 15 for controlling the entire LSI 10, and an internal bus 16 to which the DSP 14 and LSI 10 are connected. The internal bus 16 included in the LSI 10 is connected to an external memory for booting 18 and external memory for saving 19 via an external bus for booting/saving 17.

[0041] The external memory for booting 18 outside the LSI 10 stores a program and the external memory for saving 19 stores data the DSP 14 and CPU 15 handle.

[0042] If the power is turned on and off inside the LSI 10, data which the DSP 14 and CPU 15 were processing while the power was in the ON state will be lost when the power is turned off. Therefore, the data will be saved in the external memory for saving 19 before the power is turned off. When the power to the power on/off block 12 is turned on again for intermittent receiving, a program stored in the external memory for booting 18 and data saved in the external memory for saving 19 are loaded into memories for the DSP 14 and CPU 15. At this time error detection called checksum is performed on the program loaded to check whether the program was read normally. In the present invention, this checksum, will be performed at need and not each time a boot is performed.

[0043] Now, how to control whether or not this checksum is performed will be described.

[0044] Fig. 3 is a view showing the structure of checksum control.

[0045] A checksum control register 20 is located in the power on block 11 in the LSI 10 where the power is not turned on or off. This checksum control register 20 holds information indicative of control whether or not checksum is performed as a flag bit. In this case, this flag bit is defined as follows. If the flag bit is "0," then checksum is performed. If the flag bit is "1," then checksum is not performed. There is a checksum control port 21 on the DSP 14 and the checksum control register 20 and checksum control port 21 connect by a signal line 22. The checksum control register 20 is controlled via a bus outside the LSI 10 or by an internal CPU. The DSP 14 and checksum control port 21 correspond to the error check necessity judgment circuit 4 shown in Fig. 1, the DSP 14 corresponds to the error check execution circuit 5, and the checksum control register 20 corresponds to the error check necessity notification circuit 6.

[0046] Now, a boot process performed on the basis of the value of the flag bit notified by the checksum control register 20 will be described.

[0047] Fig. 4 is a flow chart showing the sequence of processes performed regarding boot programs.

[0048] First, when the power to the DSP 14 in the power on/off block 12 is turned on to perform an intermittent receiving process, the DSP 14 loads a program from the

external memory for booting 18 into a memory 23 (step S1). After the program being loaded, a port value at the checksum control port 21 connected to the checksum control register 20 is referred to and whether a checksum flag is "0" is judged (step S2). If the checksum flag is "0," then checksum is performed (step S3) and the boot process is ended. If the checksum flag is "1," then control is exercised so that the boot process will be ended without checksum being performed.

[0049] Fig. 5 is a view showing the difference between time for which the power is in the ON state in the case of checksum being performed and time for which the power is in the ON state in the case of checksum not being performed.

[0050] If checksum is performed at the time when a boot process is performed, then the power is in the ON state for the sum of time taken to load a program, time taken to perform checksum, and time taken to perform an intermittent receiving process of an intermittent receiving cycle. On the other hand, if checksum is omitted, then time of an intermittent receiving cycle for which the power is in the ON state can be shortened by time taken to perform checksum. If time for which the power is in the ON state is shortened, consumption of power attendant on a leakage current reduces. As a result, standby time and nonstop telephone call time can be lengthened.

[0051] However, this checksum is necessary for ensuring the reliability of a system and cannot completely be omitted. It therefore is preferable that control for periodically performing checksum should be exercised to maintain the reliability of a system.

[0052] Fig. 6 is a view showing the details of a checksum control register.

[0053] The checksum control register 20 includes flag bit FLAG and fields CT5 through CT0 for specifying checksum execution interval times. To improve the reliability of a system, control is exercised so that checksum will forcibly be performed one time for the number of times specified in the fields CT5 through CT0 for specifying checksum execution interval times. The power to the power on/off block 12 is turned on at the time of intermittent receiving and a boot process is performed. It is assumed that checksum execution interval times is set to, for example, 20. Then when the number of times this boot process is performed reaches 20, control is exercised so that checksum will forcibly be performed. The fields CT5 through CT0 in the checksum control register 20 correspond to the error check interval setting circuit 7 shown in Fig. 1.

[0054] Fig. 7 is a flow chart showing the sequence of processes performed regarding a boot program which performs checksum forcibly and periodically.

[0055] First, when the power to the DSP 14 in the power on/off block 12 is turned on to perform an intermittent receiving process, the DSP 14 loads a program from the external memory for booting 18 into the memory 23 (step S11). After the program being loaded, a port value

at the checksum control port 21 connected to the checksum control register 20 is referred to and whether a checksum flag is "0" is judged (step S12). If the checksum flag is "0," then checksum is performed (step S13) and the boot process is ended. If the checksum flag is "1," then whether the number of times a boot process was performed has reached a checksum execution interval times is judged (step S14). If the number of times a boot process was performed has reached the checksum execution interval times, then checksum is performed (step S13) and the boot process is ended. If the number of times a boot process was performed has not reached the checksum execution interval times, then the boot process is ended without checksum being performed.

[0056] Moreover, time needed for program load performed at the time when a boot process is performed can be shortened further. Now, a method for shortening time needed for program load will be described.

[0057] Fig. 8 is a view showing an example of dividing a program storage area in a memory.

[0058] The external memory for booting 18 which stores programs is divided into an intermittent receiving programs storage area 18a and other programs storage area 18b. The intermittent receiving programs storage area 18a stores only programs needed for an intermittent receiving process and the other programs storage area 18b stores other programs, such as one for a telephone call process.

[0059] As a result, programs needed for an intermittent receiving process need only be loaded from the intermittent receiving programs storage area 18a at the time when a boot process is performed. Therefore, compared with a case where all the programs are loaded, time of an intermittent receiving cycle for which the power is in the ON state can be shortened. This reduces consumption of power attendant on a leakage current and the life of a battery can be lengthened.

[0060] Programs stored in the other programs storage area 18b will additionally be loaded if there is an incoming call as a result of the intermittent receiving process.

[0061] Fig. 9 is a flow chart showing the sequence of processes performed regarding divided boot programs.

[0062] First, when the power to the DSP 14 in the power on/off block 12 is turned on to perform an intermittent receiving process, the DSP 14 loads a program for intermittent receiving from the intermittent receiving programs storage area 18a in the external memory for booting 18 into the memory 23 (step S21). After the program for intermittent receiving being loaded, checksum is performed to judge whether an error occurred (step S22). If an error occurred, then step S21 is repeated to reload the program for intermittent receiving. If an error is not detected by the checksum (or if checksum is omitted), then an intermittent receiving process is performed (step S23). By performing the intermittent receiving process, whether there is a local group incoming call is judged (step S24).

[0063] If there is a local group incoming call, the DSP 14 loads other programs from the other programs storage area 18b in the external memory for booting 18 (step S25). After the other programs being loaded, checksum is performed to judge whether an error occurred (step S26). If an error occurred, then step S25 is repeated to reload the other programs. If an error is not detected by the checksum (or if checksum is omitted), then local group incoming call processing is performed (step S27). Whether the result of the decoding of a message is normal is judged by this local group incoming call processing (step S28). If the result of the decoding of the message is normal, then incoming call processing is performed (S29). If the result of the decoding of the message is abnormal, then error processing is performed (step S30).

[0064] If the judgment that there is no local group incoming call is made in step S24, then data the DSP 14 and CPU 15 were processing is saved in the external memory for saving 19 (step S31) and the power is turned off.

[0065] As described above, power consumption can be reduced by omitting checksum. Even if a checksum error occurs as a result of omitting checksum, this error can be detected at the time of decoding a message. Therefore, this is not a serious impediment to operation. Moreover, when a boot is performed, the possibility that a checksum error occurs for a reason other than hardware failure is remote and does not matter.

[0066] As has been described in the foregoing, the semiconductor device according to the present invention includes an error check necessity judgment circuit for judging whether an error check must be made on data for booting loaded from an external memory at the time of a boot, and an error check will be made in accordance with a judgment made by the error check necessity judgment circuit. As a result, an error check which is made on data for booting each time intermittent operation is performed can be omitted. Therefore, time taken to perform error check processes can be shortened. That is to say, time for which the power is in the ON state can be shortened and consumption of power attendant on a leakage current can be reduced.

[0067] Furthermore, in the preferred embodiment applied to cellular telephones, only a program necessary for an intermittent receiving process is loaded at the time of a boot. This shortens time taken to load a program. Similarly, time for which the power is in the ON state can be shortened and consumption of power attendant on a leakage current can be reduced.

[0068] A reduction in consumption of power attendant on a leakage current will lengthen the lives of batteries. Moreover, smaller capacity batteries with the same life can be used, resulting in more compact, lighter portable remote terminal units.

[0069] The foregoing is considered as illustrative only of the principles of the present invention. Further, since numerous modifications and changes will readily occur

to those skilled in the art, it is not desired to limit the invention to the exact construction and applications shown and described, and accordingly, all suitable modifications and equivalents may be regarded as falling within the scope of the invention in the appended claims.

Claims

1. A semiconductor device with an intermittent operation block where application and disconnection of power are performed periodically and repeatedly, the semiconductor device comprising:
 - an error check necessity judgment circuit for judging whether an error check must be made on data for booting loaded from an external memory by a boot process performed at the time of power being applied; and
 - an error check execution circuit for making an error check on the data for booting in the case of the error check necessity judgment circuit judging that an error check must be made.
2. The semiconductor device according to claim 1, further comprising an input terminal for accepting error check necessity data to give instructions from the outside about whether an error check must be made, wherein the error check necessity judgment circuit judges on the error check necessity data of which the error check necessity judgment circuit is notified from the outside whether an error check must be made.
3. The semiconductor device according to claim 1 or 2, further comprising an error check necessity notification circuit for notifying the error check necessity judgment circuit of error check necessity data to give instructions about whether an error check must be made.
4. The semiconductor device according to claim 3, wherein the error check necessity notification circuit is located in a block where power is not disconnected.
5. The semiconductor device according to claim 3 or 4, wherein the error check necessity notification circuit is a control register which stores a flag bit indicative of whether an error check must be made.
6. The semiconductor device according to claim 3, 4 or 5, further comprising an error check interval setting circuit for forcibly setting error check necessity data of which the error check necessity notification circuit notifies the error check necessity judgment circuit to data indicative that an error check must be made only at the time of the number of times a boot is performed reaching a certain number.
7. The semiconductor device according to claim 6, wherein the error check interval setting circuit is located in a block where power is not disconnected.
8. The semiconductor device according to claim 6 or 7, wherein the error check interval setting circuit can arbitrarily set the number of times the boot is performed, being a cycle in which an error check is made.
9. A portable remote terminal unit in which power to blocks unnecessary for an intermittent receiving control process is disconnected for a period when the unit in a standby state does not perform intermittent receiving, the portable remote terminal unit comprising:
 - an error check necessity judgment circuit for judging whether an error check must be made on data for booting loaded from a memory where power is not disconnected by a boot process performed at the time of power being applied to perform the intermittent receiving;
 - an error check execution circuit for making an error check on the data for booting in the case of the error check necessity judgment circuit judging that an error check must be made;
 - an error check necessity notification circuit for notifying the error check necessity judgment circuit whether an error check must be made; and
 - an error check interval setting circuit for causing the error check necessity notification circuit to notify the error check necessity judgment circuit that an error check must be made at the time of the number of times a boot is performed reaching a certain number.
10. The portable remote terminal unit according to claim 9, wherein the error check necessity notification circuit and the error check interval setting circuit are located in a block where power is not disconnected.
11. The portable remote terminal unit according to claim 9 or 10, wherein the error check interval setting circuit can arbitrarily set the number of times the boot is performed, being a cycle in which an error check is made.
12. An intermittent receiving method in which power to blocks in a portable remote terminal unit unnecessary for an intermittent receiving control process is disconnected for a period when the unit in a standby state does not perform intermittent receiving, the intermittent receiving method comprising the steps of:

applying power to blocks in the portable remote terminal unit necessary for an intermittent receiving control process at the time of intermittent receiving;

loading only a program necessary for the intermittent receiving control process from an external memory; 5

judging whether the number of times a boot is performed has reached a certain number; and making an error check on the program loaded at the time of the number of times the boot is performed reaching a certain number. 10

13. The intermittent receiving method according to claim 12, further comprising the step of loading remaining programs from the memory in the case of there being a local group incoming call at the time of the intermittent receiving control process. 15

14. A portable remote terminal unit in which power to blocks unnecessary for an intermittent receiving control process is disconnected for a period when the unit in a standby state does not perform intermittent receiving, the portable remote terminal unit comprising: 20

an error check necessity judgment circuit for judging whether an error check must be made on data for booting loaded from a memory where power is not disconnected by a boot process performed at the time of power being applied to perform the intermittent receiving; and 30

an error check execution circuit for making an error check on the data for booting in the case of the error check necessity judgment circuit judging that an error check must be made. 35

40

45

50

55

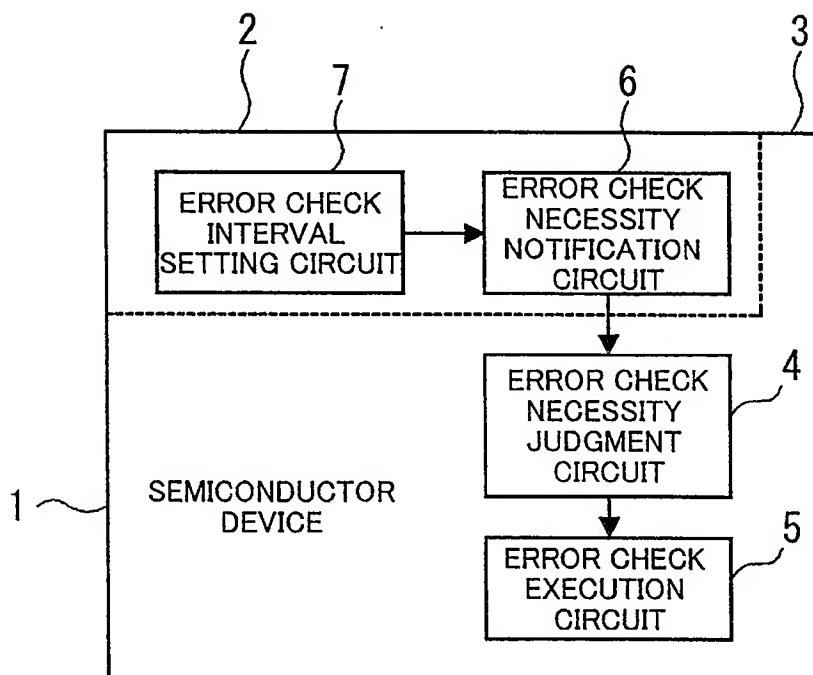


FIG. 1

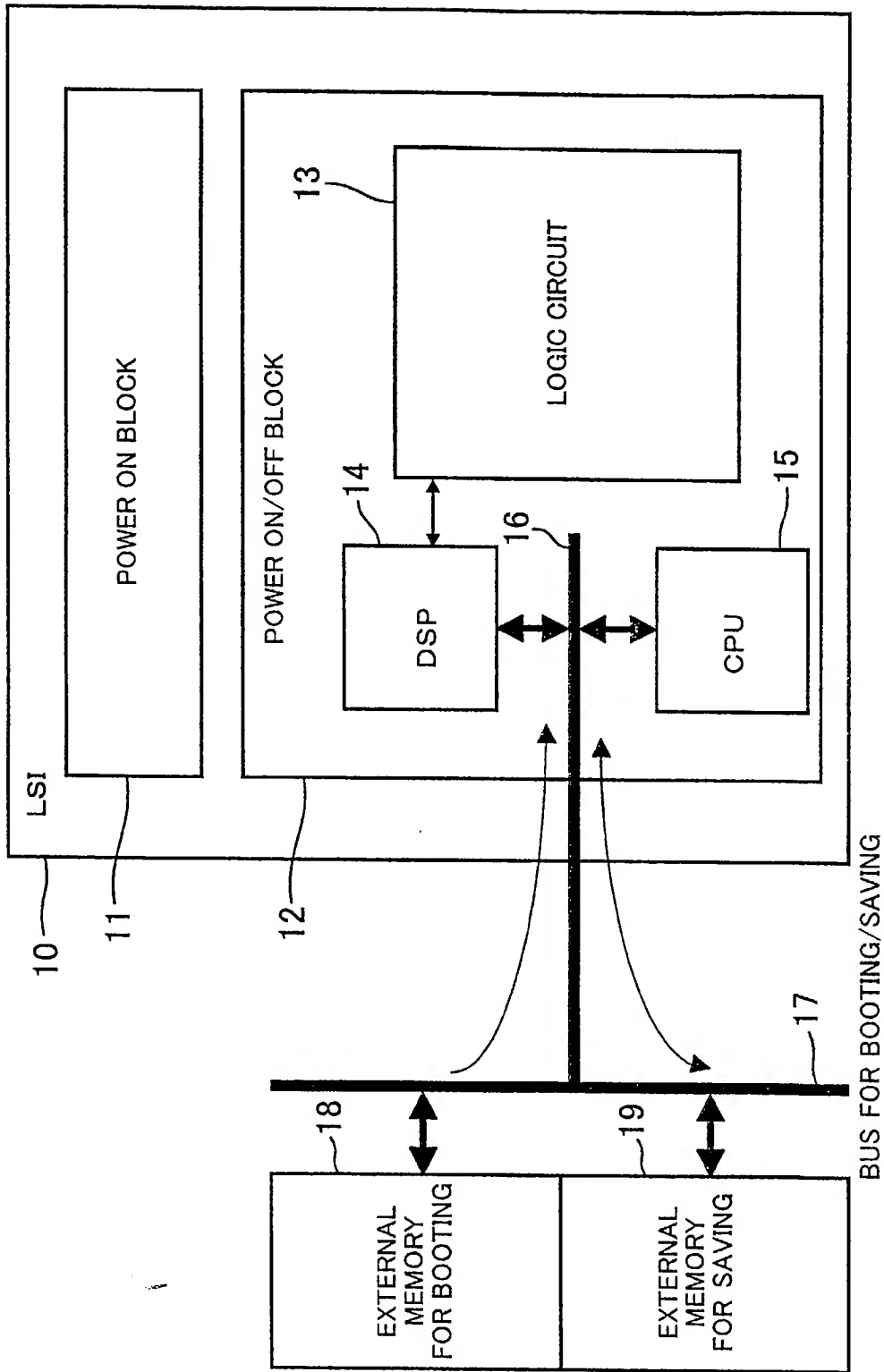


FIG. 2

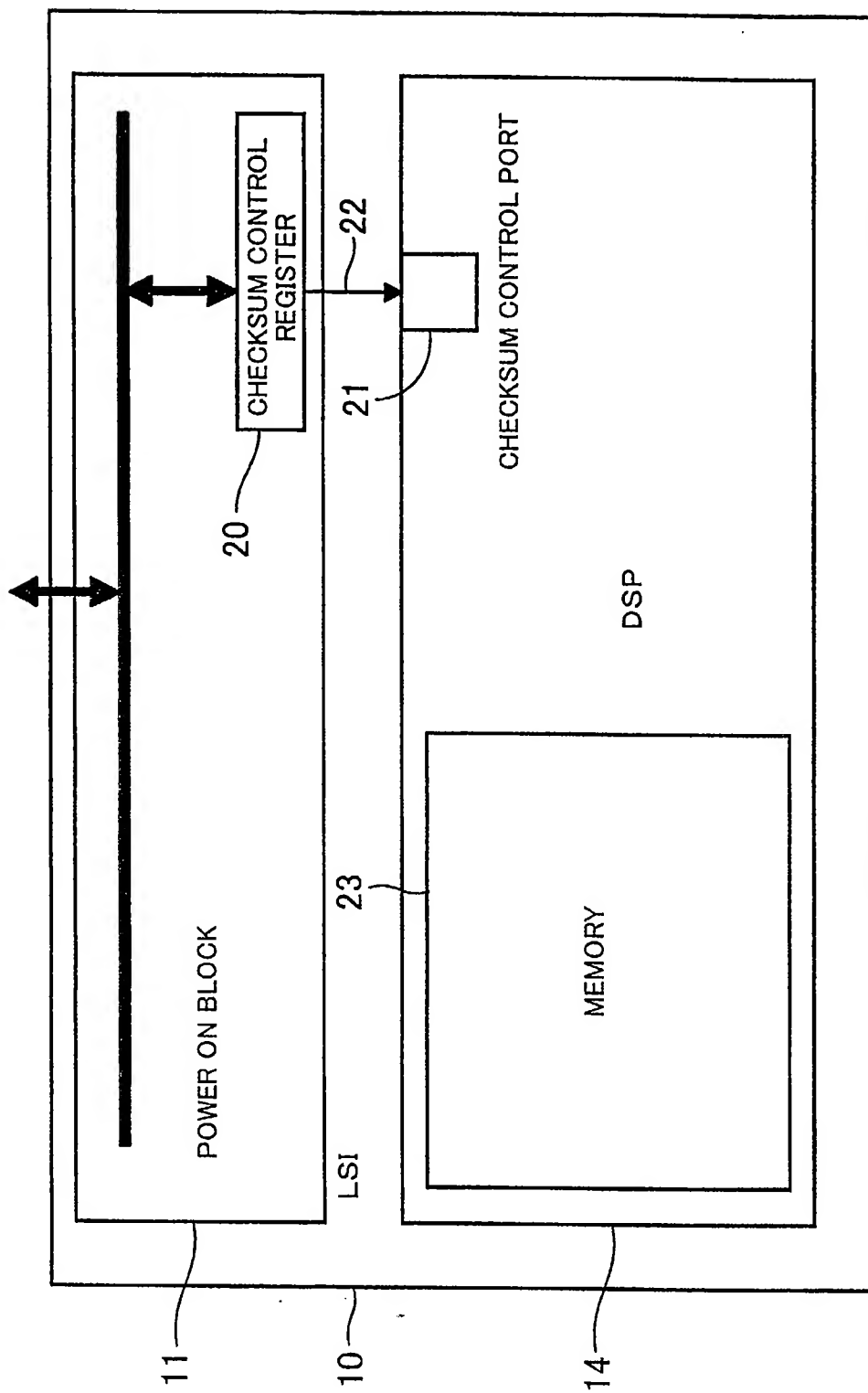


FIG. 3

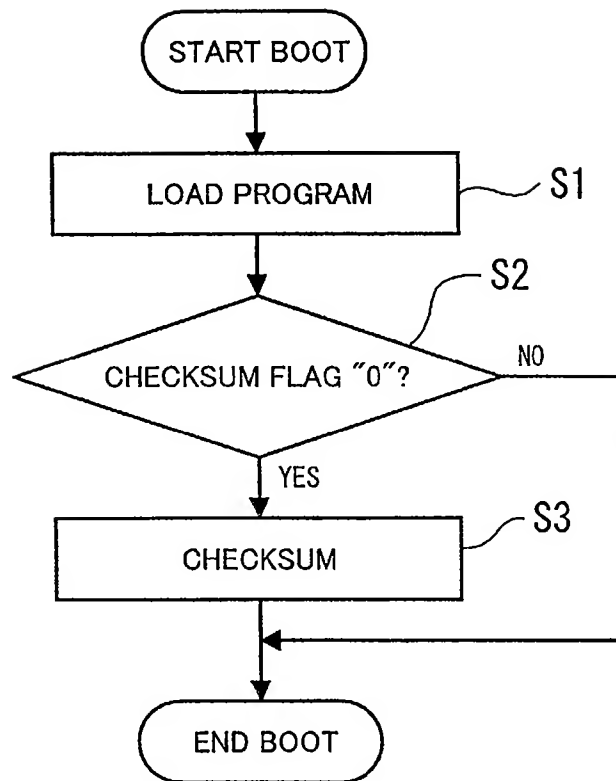


FIG. 4

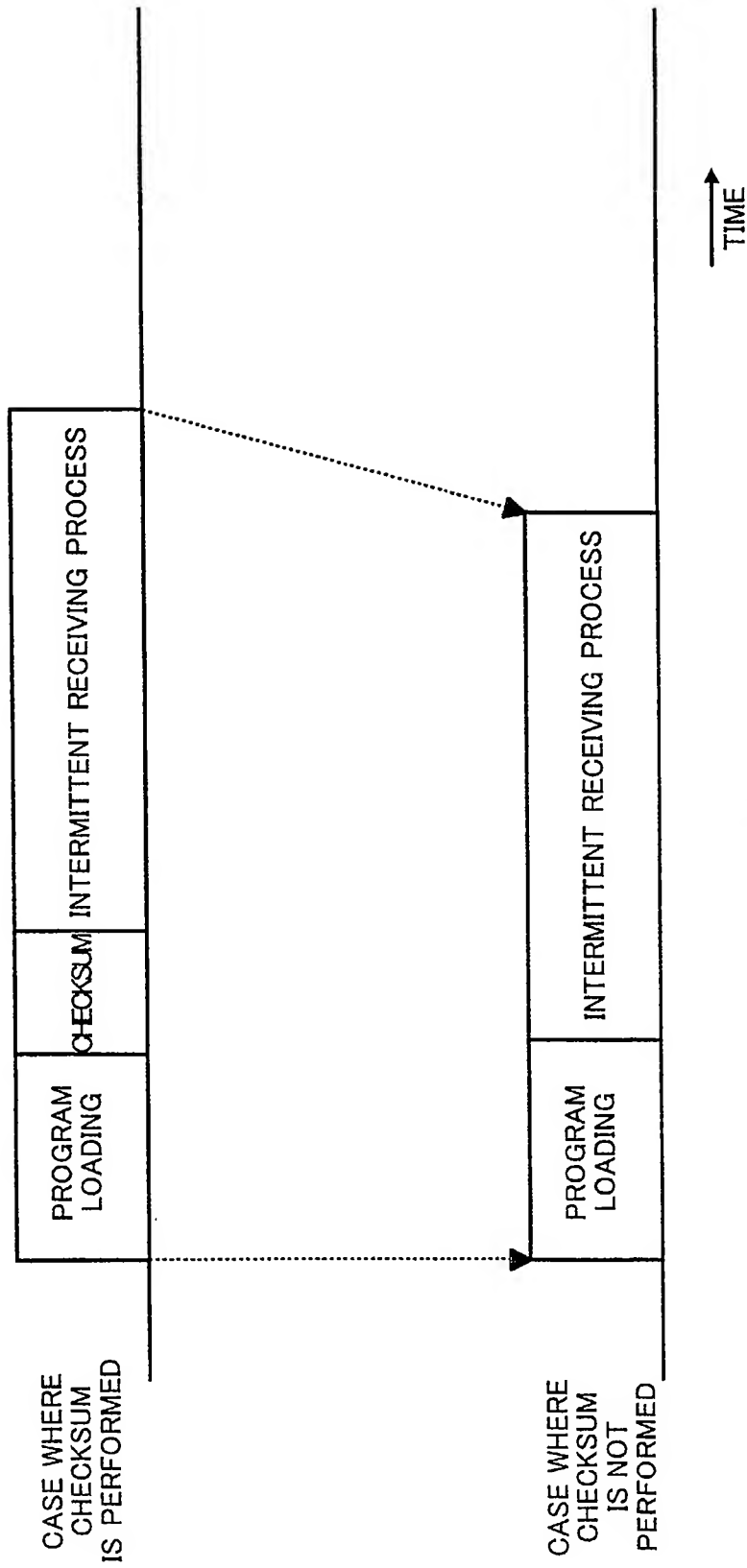
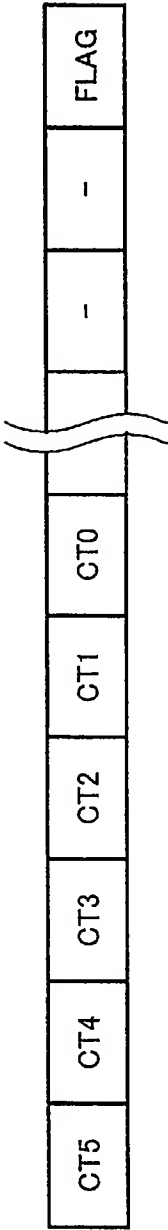


FIG. 5



CT5~0 : SPECIFY CHECKSUM EXECUTION INTERVAL TIMES

FLAG : FLAG INDICATIVE OF WHETHER CHECKSUM IS PERFORMED

- 0: CHECKSUM IS PERFORMED
- 1: CHECKSUM IS NOT PERFORMED

FIG. 6

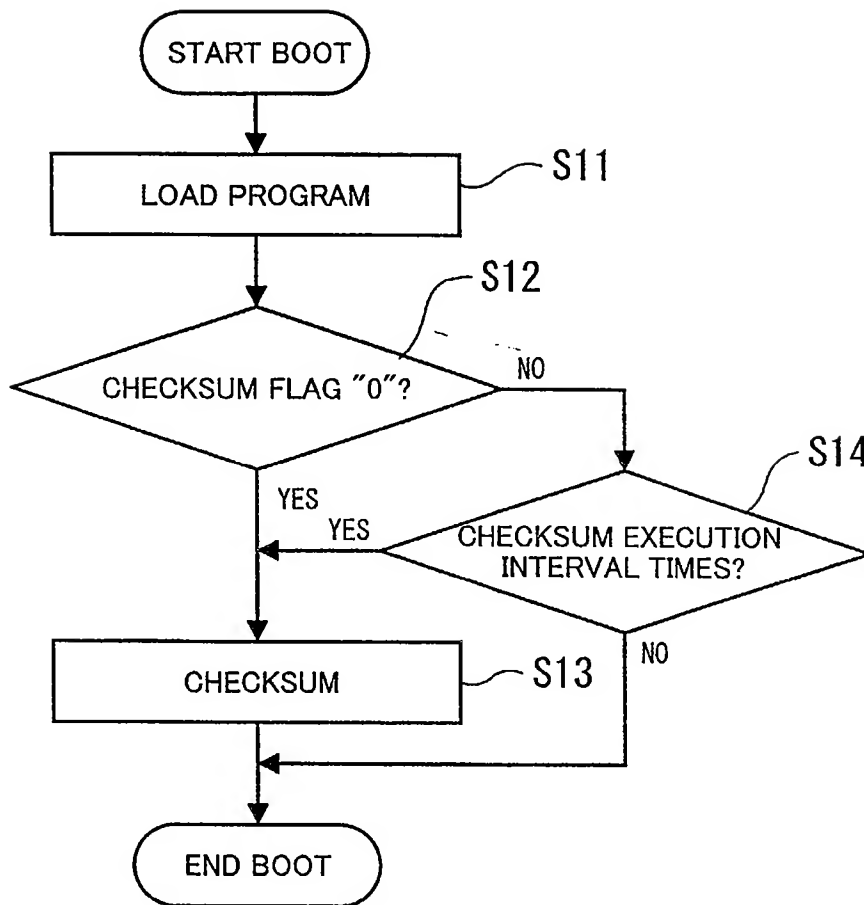


FIG. 7

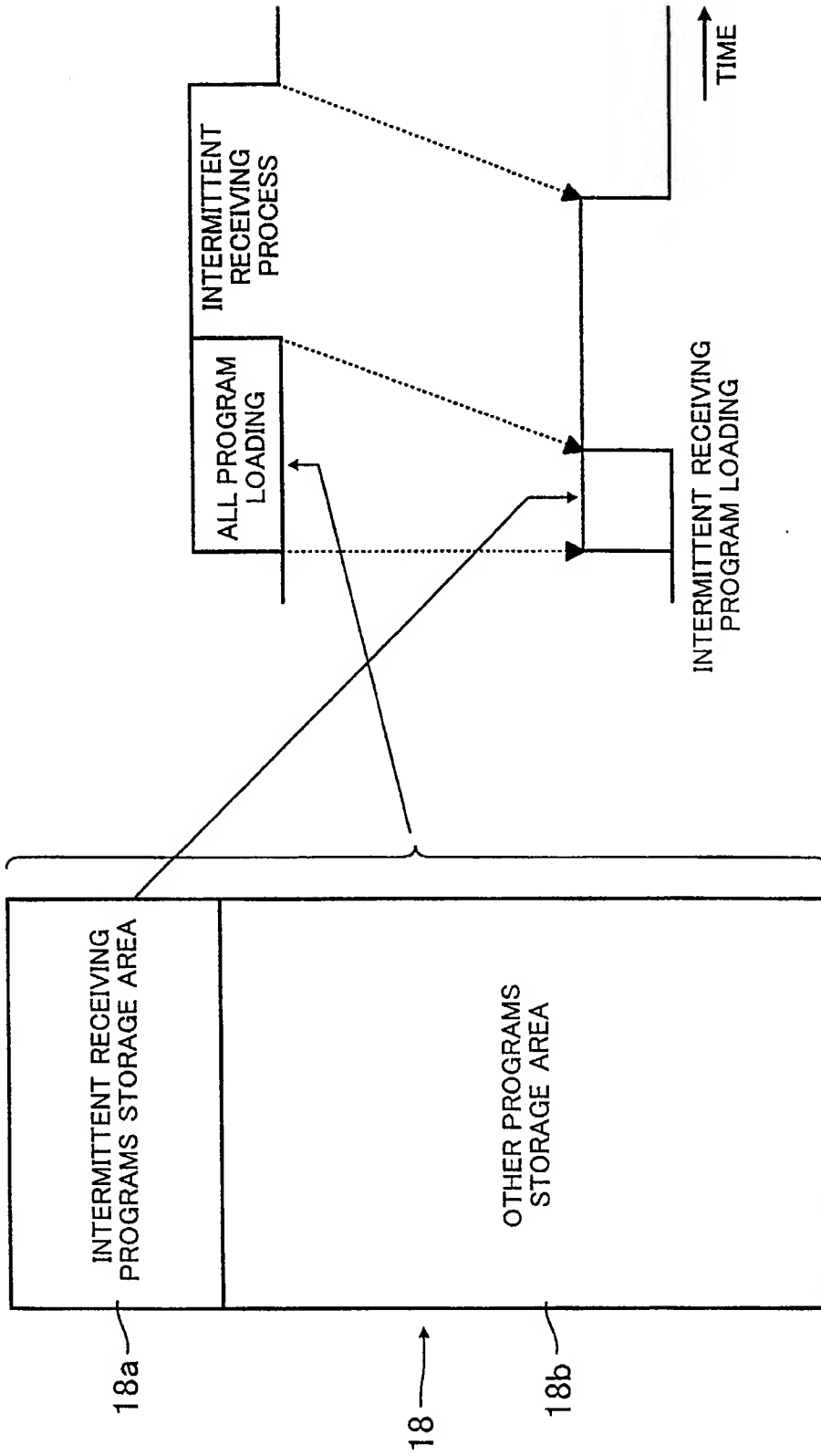


FIG. 8

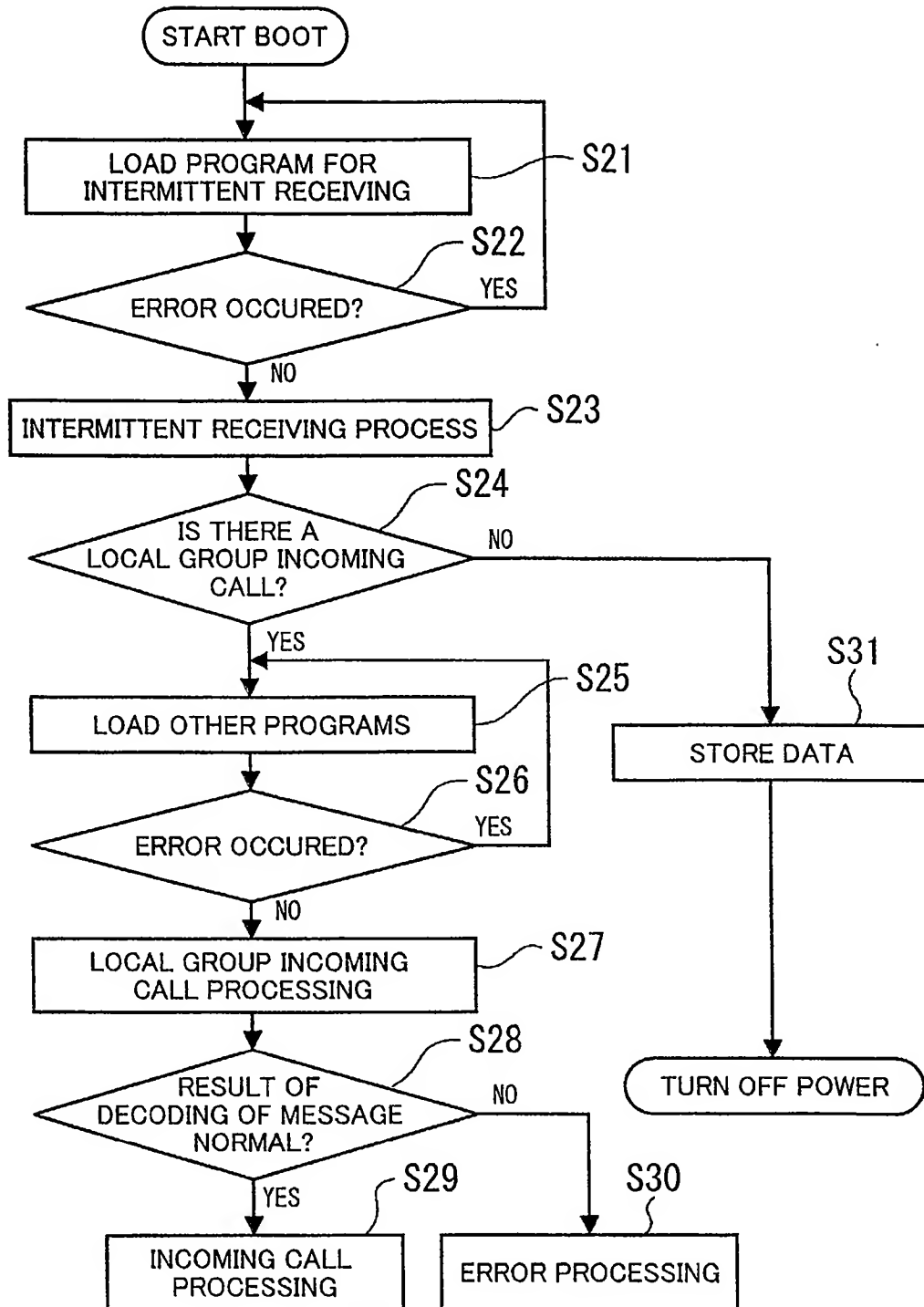


FIG. 9

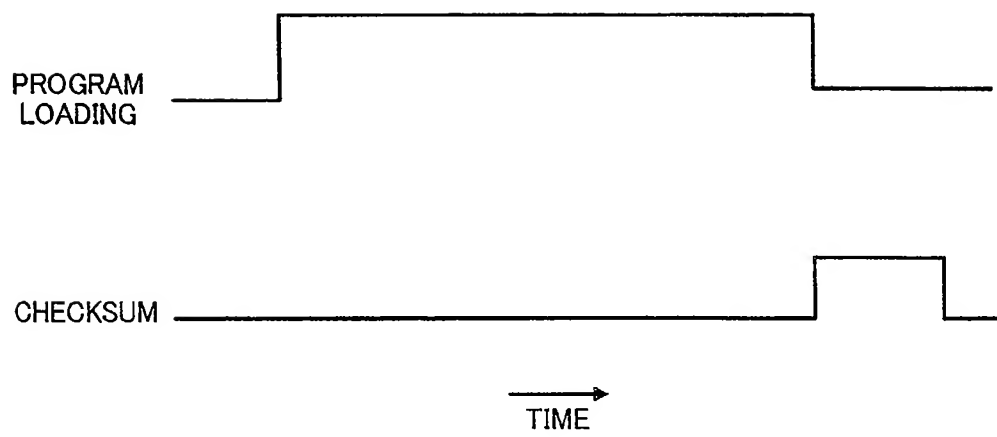


FIG.10

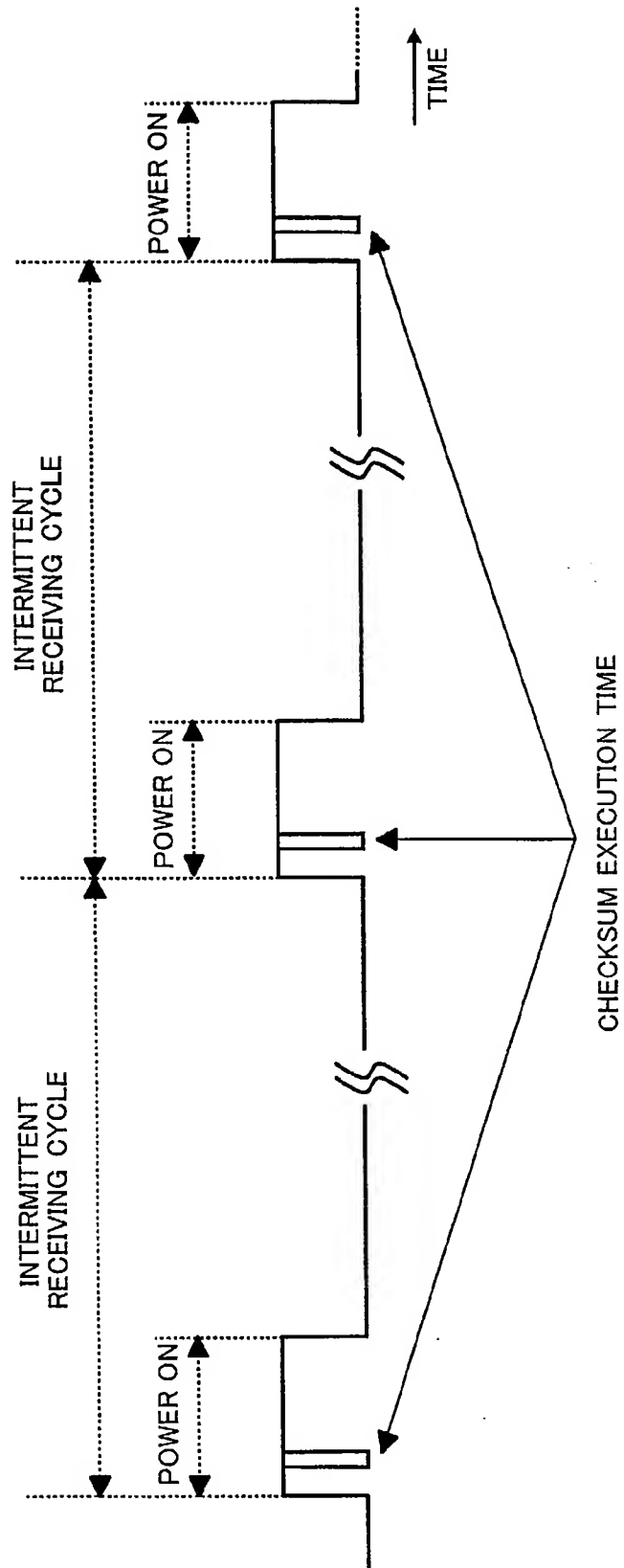


FIG. 11

